UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/607,167	06/27/2003	Hyong-Kyun Lee	P56833	5599

7590   05/02/2007

Robert E. Bushnell
Suite 300
1522 K Street, N.W.
Washington, DC 20005

EXAMINER
ANYA, CHARLES E

ART UNIT	PAPER NUMBER
2194	

MAIL DATE	DELIVERY MODE
05/02/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

	Application No.	Applicant(s)
Office Action Summary	10/607,167	LEE, HYONG-KYUN
	Examiner	Art Unit
	Charles E. Anya	2194

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE _3_ MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.
- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
  Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on _02 February 2007_.

2a)☒ This action is **FINAL.**   2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) _1-22_ is/are pending in the application.

    4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☒ Claim(s) _6-11_ is/are allowed.

6)☒ Claim(s) _1-3 and 12-21_ is/are rejected.

7)☒ Claim(s) _4,5 and 22_ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

    Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

    Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

    a)☒ All   b)☐ Some * c)☐ None of:

      1.☐ Certified copies of the priority documents have been received.

      2.☐ Certified copies of the priority documents have been received in Application No. _____.

      3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

    * See the attached detailed Office action for a list of the certified copies not received.

WILLIAM THOMSON
SUPERVISORY PATENT EXAMINER

Attachment(s)

1)☐ Notice of References Cited (PTO-892)

2)☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3)☒ Information Disclosure Statement(s) (PTO/SB/08)
    Paper No(s)/Mail Date _3/1/05; 12/13/06_.

4)☐ Interview Summary (PTO-413)
    Paper No(s)/Mail Date. _____ .

5)☐ Notice of Informal Patent Application

6)☐ Other: _____.

DETAILED ACTION

1.   Claims 1-22 are pending in this application.

Claim Rejections - 35 USC § 102

2.   The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that

form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

3.   Claim 3 is rejected under 35 U.S.C. 102(b) as being anticipated by U.S. Pat.

No. 6,223,134 B1 to Rust et al.

4.   As to claim 3, Rust teaches a method for commonly controlling device drivers,

comprising the steps of: arranging a device independent access hierarchy between an

application hierarchy and a device driver hierarchy (figure 5 (Class Drivers 304) Col. 13

Ln. 19 – 67); defining functions available in a corresponding device driver among

functions of a function block in a function table (Col. 6 Ln. 34 – 51, IVI engine 306 Col.

16 Ln. 28 – 49, "...array of function names..." Col. 23 Ln. 29 – 42); when a device is

initialized, allowing said device independent access hierarchy to generate a device

handler identifier based on a standardized data format for said device and transmit the

generated device handler identifier to the application hierarchy of a higher order

("...returns a handle..." Col. 20 Ln. 35 – 46, "...handle...returned..." Col. 23 Ln. 12 – 21,

"...returns a handle..." Col. 24 Ln. 22 – 29); and allowing the higher-order application

hierarchy to call a predetermined device using the device handler identifier ("...this

handle..." Col. 23 Ln. 12 – 28, "...use this handle..." Col. 24 Ln. 22 – 29), and allowing

said device independent access hierarchy to identify a function of the corresponding

device driver from the function table using the device handler identifier ("...pointer..."

Col. 16 Ln. 28 – 49, "...this handle..." Col. 23 Ln. 12 – 28, Col. 24 Ln. 1 – 11) and call

the function of the corresponding device driver ("...this handle..." Col. 23 Ln. 12 – 28,

Col. 24 Ln. 22 – 29).

Claim Rejections - 35 USC § 103

5.   The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all

obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set
> forth in section 102 of this title, if the differences between the subject matter sought to be patented and
> the prior art are such that the subject matter as a whole would have been obvious at the time the
> invention was made to a person having ordinary skill in the art to which said subject matter pertains.
> Patentability shall not be negatived by the manner in which the invention was made.

6.   Claims 1,2 and 12-21 are rejected under 35 U.S.C. 103(a) as being

unpatentable over U.S. Pat. No. 6,223,134 B1 to Rust et al. in view of U.S. Pat. No.

6,993,772 B2 to Pike et al.

7.   As to claim 1, Rust teaches a method for commonly controlling device drivers,

comprising the steps of: arranging a device independent access hierarchy between an

application hierarchy and a device driver hierarchy (figure 5 (Class Drivers 304) Col. 13

Ln. 19 – 67) and applying a standardized rule of said device independent access

hierarchy to said application hierarchy and said device driver hierarchy (IVI Engine 306

Col. 16 Ln. 28 – 49); and allowing said application hierarchy to access the device driver

hierarchy through the standardized rule of said device independent access hierarchy

(Col. 6 Ln. 34 – 51, Class Drivers 304 Col. 16 Ln. 28 – 67).

Rust is silent with reference to allowing said device driver hierarchy to access

said application hierarchy through the standardized rule of said device independent

access hierarchy.

Pike teaches allowing said device driver hierarchy to access said application

hierarchy through the standardized rule of said device independent access hierarchy

(Instrument Engine 102 ("... sent and read to and from..." Col. 8 Ln. 3 – 14).

It would have been obvious to one of ordinary skill in the art at the time the

invention was made to modify system of Rust with the teaching of Pike because the

teaching of Pike would improve the system of Rust by allowing communication between

a user application and device driver such that a response could be returned to the user

application.

8.   As to claim 2, Pike teaches the method as set forth in claim 1, with said step of

allowing said application hierarchy and said device driver hierarchy to access,

comprising the steps of: allowing said application hierarchy to transmit control

commands based on a standardized common format for a corresponding device driver

to said device independent access hierarchy, and allowing said device independent

access hierarchy to convert the control commands into other control commands based

on a local format and transmit the converted control commands to said device driver;

and allowing said device driver to give a response to the converted control commands

based on the local format to said device independent access hierarchy, and allowing the

device independent access hierarchy to convert the response from said device driver

into a response based on the standardized common format and transmit the response

based on the standardized common format to said application hierarchy

("...translate(s)..." Col. 7 Ln. 25 – 55, Instrument Engine 102 ("...formats...sent and

read to and from..." Col. 8 Ln. 3 – 14).

9.   As to claim 12, Rust teaches a method, comprising: requesting loss of signal

("...check instrument status...") state information based on a standardized common

format by an application to a device independent access hierarchy (figure 8A (Step 472)

Col. 24 Ln. 32 – 37); the device independent access hierarchy allows the request from

said application to provided in a first device local format ("...only required to have

knowledge of the class driver 304..." Col. 15 Ln. 24 – 45) and requesting a first device

driver to provide the loss of signal state information to said device independent access

hierarchy (figure 8A (step 478) Col. 24 Ln. 43 – 45); responding to the request for loss

of signal state information based on the first device local format (figure 8B (Step 487)

Col. 23 – 27); responding to said application by said device independent access

hierarchy for loss of signal state information based on the standardized common format

(Check Status Callback Col. 36 Ln. 46 – 49, figure 25B (Step 928) Col. 40 Ln. 15 - 20).

Rust does not explicitly teach converting the request from said application into a

first device local format.

Pike teaches converting the request from said application into a first device local

format ("...translate(s)..." Col. 7 Ln. 25 – 55, Instrument Engine 102 ("...formats...sent

and read to and from..." Col. 8 Ln. 3 – 14).

It would have been obvious to one of ordinary skill in the art at the time the

invention was made to modify the system of Rust with the teaching of Pike because the

teaching of Pike would improve the system of Rust by providing a powerful, concise

mechanism that allows users to easily and quickly communicate details of different

hardware interfaces having a distinct application program interface or driver modules

(Pike Col. 3 Ln. 31 – 35).

10.   As to claim 13, Rust teaches the method of claim 12, with said step of converting

the request from said application further comprising of converting the request into a

second device local format and requesting a second device driver to provide the loss of

signal state information to said device independent access hierarchy based on the

second device local format when a first device is converted to a second device and said

first device driver is changed to said second device driver (Abstract ("...interchangeable

instrument drivers), "...interchangeability..." Col. 6 Ln. 18 – 28, "...substitute other

instruments..." Col. 7 Ln. 42 – 50, "...replace..." Col. 8 Ln. 22 – 28, Col. 15 Ln. 3 - 6).

11.   As to claim 14, Rust teaches the method of claim 13, further comprising of

converting control commands based on the standardized common format to control

commands provided to the device drivers accommodating a change of said application

to a second application without changing the control commands provided to the device

drivers (Abstract "...interchangeable instrument drivers...").

12.   As to claim 15, Rust teaches the method of claim 14, further comprised of

providing a mutual interface between said application and said first and second device

drivers by the device independent access hierarchy (Col. 13 Ln. 19 – 44); reading

material from a device driver control block and accessing the first and second device

drivers using predetermined functions (figure 5, Col. 16 Ln. 28 – 56, Col. 20 Ln. 1 – 46).

13.   As to claim 16, Rust teaches the method of claim 15, further comprising of said

device independent access hierarchy using device handler identifiers based on the

standardized data format, said device handler identifiers corresponding to respective

devices ("...handle..." Col. 20 Ln. 38 – 46, "...handle..." Col. 23 Ln. 12 – 21).

14.   As to claim 17, Rust teaches the method of claim 16, further comprising:

providing the device handler identifiers to said application from said device independent

access hierarchy during an initialization of the corresponding device; and storing, by

said application, the device handler identifiers and calling a corresponding device using

a corresponding device handler identifier ("...handle..." Col. 20 Ln. 38 – 46,

"...handle..." Col. 23 Ln. 12 – 21).

15.   As to claim 18, Rust teaches the method of claim 17, further comprising of said

device independent access hierarchy determining according to said device handler

identifier whether a certain device driver should be called and calling the certain device

handler according to the determination ("...handle..." Col. 20 Ln. 38 – 46, "...handle..."

Col. 23 Ln. 12 – 21).

16.   As to claim 19, Rust the method of claim 18, with the device independent access

hierarchy using certain pointers and function pointers in performing the standardized

common format in the device independent access hierarchy (Col. 16 Ln. 22 – 49, Col.

20 Ln. 52 – 67, Col. 21 Ln. 17 – 30).

17.   As to claim 20, Rust teaches the method of claim 19, further comprised of when

said application is calling a function of a function block to be used, said device

independent access hierarchy identifies the existence of a corresponding function from

a function table and uses a device handler identifier to inform the initialization of the

device driver accommodating said application to access a device driver using said

device handler identifier ("...handle..." Col. 20 Ln. 38 – 46, "...handle..." Col. 23 Ln. 12

– 21).

18.   As to claim 21, Rust teaches the method of claim 20, further comprised of not

varying the device handler identifier value for the device when said first device driver is

changed to said second device driver ("...handle..." Col. 20 Ln. 38 – 46, "...handle..."

Col. 23 Ln. 12 – 21).

Allowable Subject Matter

Claims 6-11 are allowed.

Claims 4-5 and 22 are objected to as being dependent upon a rejected base

claim, but would be allowable if rewritten in independent form including all of the

limitations of the base claim and any intervening claims.

Response to Arguments

Applicant's arguments filed 2/2/07 have been fully considered but they are not

persuasive, however, Examiner acknowledges Applicant argument regarding claims 4

and 6 and has as result objected to claims 4,5, and 22 and allowed claim 6-11.

Applicant argues in substance that (1) the Rust prior art does not teach the terms

"device independent access", "device independent hierarchy", "application hierarchy"

and "device driver hierarchy", (2) the Rust prior art does not teach define functions

available in a corresponding device driver among functions of a function block in a

function table, (3) the Rust prior art does not teach a device handler identifier generated

based on standardized data format for the device, (4) it is not obvious to combine the

Rust and Pike prior arts, and (5) the Rust prior art does not teach loss of signal state of

information.

As to point (1), although the Rust prior art does not explicitly teach the listed

terms, the Rust prior art does teach the same function as the listed terms. The test of

whether a reference teaches the language of a claimed invention is not whether the

terms used in the claims are the same as those used in the reference, but whether their

functions as the same. In claim 3, the "device independent access hierarchy" is

arranged between an application hierarchy and a device driver hierarchy, functions to

generate a device handler identifier, transmitting the device handler identifier to the

application hierarchy and identifying a function of a corresponding device driver. The

Rust prior art teaches these limitations by providing a class driver that is sandwiched by

user applications and specific drivers (figure 5). The Class driver functions to retrieve

function pointers from the IVI Engine 306, as a result identifying a function of the

specific driver and allowing the user application to use the function pointer to call a

corresponding function of the specific driver (Col. 6 Ln. 19 – 43).

This notwithstanding the figures 4,5 and 26 discloses a hierarchy of applications,

device drivers and devices.

As to point (2), this limitation as claimed discloses defined functions that are

associated with device drivers and the function are organized in a table. Although the

Rust prior art does not explicitly describe a table, it does disclose defined functions

provided in the specific drivers that are callable by user applications (Col. 6 Ln. 19 – 51,

Col. 14 Ln. 45 – 55) and the defined function is found by the IVI Engine 306 by looking

in the specific driver (Col. 14 Ln. 52 – 54, Col. 21 Ln. 24 – 30). The fact that these defined functions could be looked up in the specific driver implies that there has to be location in the specific driver where they are stored.

As to point (3), the function pointer received from the IVI engine 306 has to correspond to the standard data format for the specific device otherwise how would the user application communicate with the specific device or instrument.

As to point (4), the test for obviousness is not whether the features of a secondary reference may be bodily incorporated into the structure of the primary reference; nor is it that the claimed invention must be expressly suggested in any one or all of the references. Rather, the test is what the combined teachings of the references would have suggested to those of ordinary skill in the art. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981).

As to point (5), as claimed "requesting loss of signal state information" is a request made by an application for indication of error with device. The utility function in the specific driver provides this function by allowing an application to make call to the specific driver via the utility function for instrument/device state/status and when an error is encountered a message is returned (Col. 26 Ln. 24 – 33, Col. 36 Ln. 46 – 49).

Conclusion

THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE

MONTHS from the mailing date of this action. In the event a first reply is filed within

TWO MONTHS of the mailing date of this final action and the advisory action is not

mailed until after the end of the THREE-MONTH shortened statutory period, then the

shortened statutory period will expire on the date the advisory action is mailed, and any

extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of

the advisory action. In no event, however, will the statutory period for reply expire later

than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the

examiner should be directed to Charles E. Anya whose telephone number is (571) 272-

3757. The examiner can normally be reached on M-F (8:30-5:00).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's

supervisor, William Thomson can be reached on (571) 272-3718. The fax phone

number for the organization where this application or proceeding is assigned is 571-

273-8300.

Information regarding the status of an application may be obtained from the

Patent Application Information Retrieval (PAIR) system. Status information for

published applications may be obtained from either Private PAIR or Public PAIR.

Status information for unpublished applications is available through Private PAIR only.

For more information about the PAIR system, see http://pair-direct.uspto.gov. Should

you have questions on access to the Private PAIR system, contact the Electronic

Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a

USPTO Customer Service Representative or access to the automated information

system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Charles E Anya
Examiner
Art Unit 2194

cea.

WILLIAM THOMSON
SUPERVISORY PATENT EXAMINER